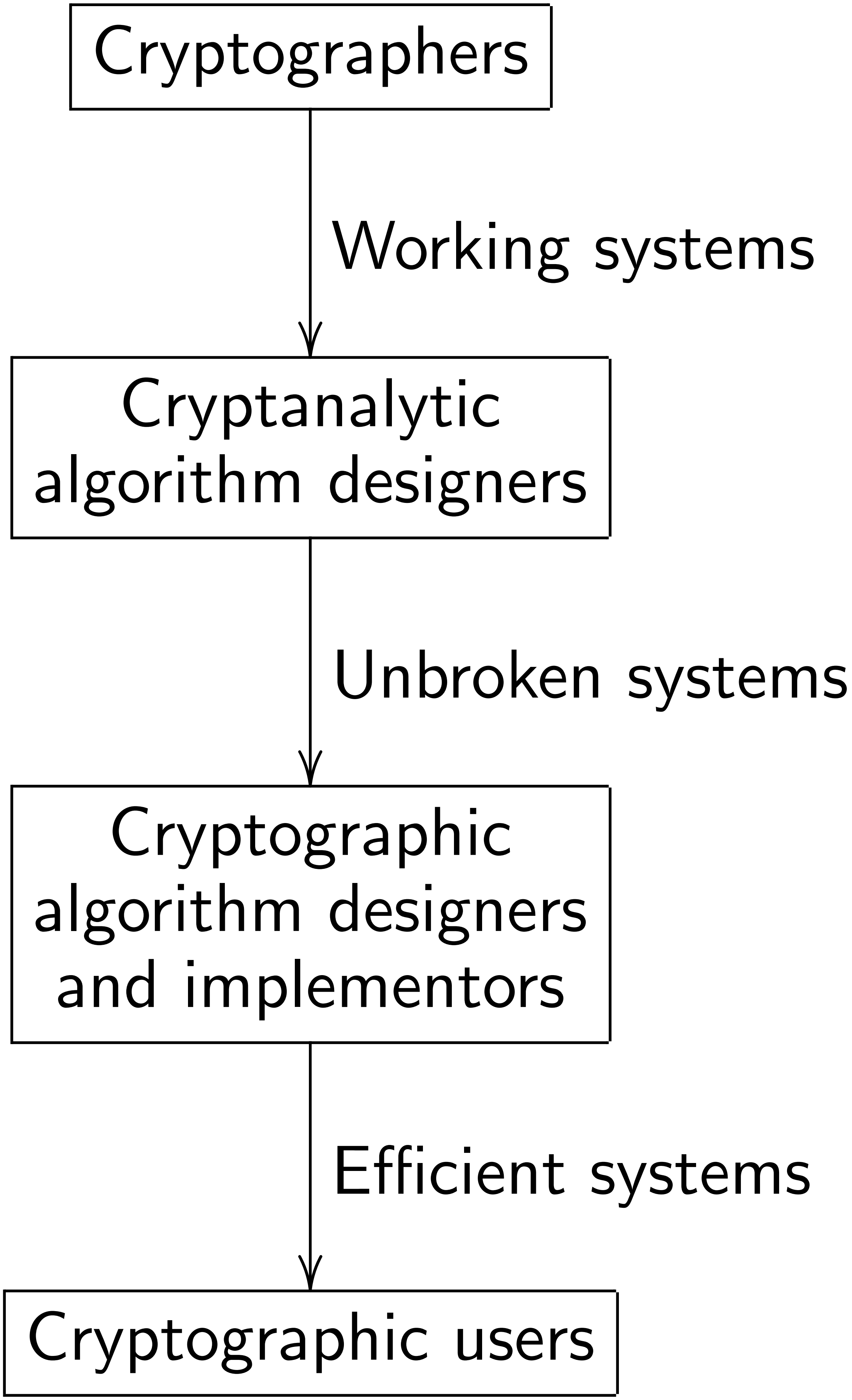
Post-quantum cryptography

D. J. Bernstein
University of Illinois at Chicago,
Technische Universiteit Eindhoven


Cryptographers
Working systems
Cryptanalytic algorithm designers
Unbroken systems
Cryptographic algorithm designers and implementors
Efficient systems
Cryptographic users

1. Working systems

Fundamental question for cryptographers:
How can we encrypt, decrypt, sign, verify, etc.?

Many answers:
DES, Triple DES, FEAL-4, AES, RSA, McEliece encryption, Merkle hash-tree signatures, Merkle–Hellman knapsack encryption, Buchmann–Williams class-group encryption, ECDSA, HFE^{v-}, NTRU, et al.

Detailed example
(not a very good cryptosystem!):
textbook exponent-3 RSA-1024.

Receiver's secret key: distinct 512-bit primes $p, q \in 2 + 3\mathbf{Z}$.

Receiver's public key: pq.

Sender's plaintext:
$m \in \{0, 1, \ldots, pq - 1\}$.

Sender's ciphertext: $m^3 \bmod pq$.

Receiver uses p, q to compute m given $m^3 \bmod pq$.

2. Unbroken systems

Fundamental question for *pre-quantum* cryptanalysts: What can an attacker do using $<2^b$ operations on a *classical* computer?

Fundamental question for *post-quantum* cryptanalysts: What can an attacker do using $<2^b$ operations on a *quantum* computer?

Goal: identify systems that are *not* breakable in $<2^b$ operations.

Examples of RSA cryptanalysis:

Schroeppel's "linear sieve", mentioned in 1978 RSA paper, factors pq into p, q using $(2+o(1))^{(\lg pq)^{1/2}(\lg \lg pq)^{1/2}}$ simple operations (conjecturally).

To push this beyond 2^b, must choose pq to have at least $(0.5+o(1))b^2/\lg b$ bits.

Note 1: $\lg = \log_2$.

Note 2: $o(1)$ says *nothing* about, e.g., $b = 128$.

1993 Buhler–Lenstra–Pomerance, generalizing 1988 Pollard "number-field sieve", factors pq into p, q using $(3.79\ldots + o(1))^{(\lg pq)^{1/3}(\lg \lg pq)^{2/3}}$ simple operations (conjecturally).

To push this beyond 2^b, must choose pq to have at least $(0.015\ldots + o(1))b^3/(\lg b)^2$ bits.

Subsequent improvements: $3.73\ldots$; details of $o(1)$.
But can reasonably conjecture that $2^{(\lg pq)^{1/3+o(1)}}$ is optimal —for classical computers.

Many "protocol" attacks.

e.g. attacker guesses user's m, verifies m^3 mod pq.

e.g. attacker hopes $m < (pq)^{1/3}$.

e.g. attacker sees how receiver reacts to $8m^3$ mod pq.

Typical fix: feed m through randomization+padding+"AONT".

"Simple RSA" (2001 Shoup): send r^3 mod pq for random r; use hash of r as AES-GCM key to encrypt and authenticate m.

Cryptographic systems surviving *pre-quantum* cryptanalysis:

Triple DES (for $b \le 112$),
AES-256 (for $b \le 256$),
RSA with $b^{3+o(1)}$-bit modulus,
McEliece with code length $b^{1+o(1)}$, Merkle signatures with "strong" $b^{1+o(1)}$-bit hash,
BW with "strong" $b^{2+o(1)}$-bit discriminant, ECDSA with "strong" $b^{1+o(1)}$-bit curve,
HFE^{v-} with $b^{1+o(1)}$ polynomials,
NTRU with $b^{1+o(1)}$ bits, et al.

Typical algorithmic tools for
pre-quantum cryptanalysts:
NFS, ρ, ISD, LLL, F4, XL, et al.

Post-quantum cryptanalysts
have all the same tools
plus quantum algorithms.

Spectacular example:
1994 Shor factors pq into p, q
using $(\lg pq)^{2+o(1)}$
simple quantum operations.
To push this beyond 2^b,
must choose pq to have at least
$2^{(0.5+o(1))b}$ bits. Yikes.

Cryptographic systems surviving *post-quantum* cryptanalysis:

AES-256 (for $b \le 128$),
McEliece code-based encryption with code length $b^{1+o(1)}$,
Merkle hash-based signatures with "strong" $b^{1+o(1)}$-bit hash,
$\text{HFE}^{\text{v}-}$ MQ signatures with $b^{1+o(1)}$ polynomials,
NTRU lattice-based encryption with $b^{1+o(1)}$ bits,
et al.

3. Efficient systems

Fundamental question for designers and implementors of cryptographic algorithms: Exactly how efficient are the unbroken cryptosystems?

Many goals: minimize encryption time, size, decryption time, etc.

Pre-quantum example:
ECDSA with "strong" $b^{1+o(1)}$-bit curve verifies signature in $b^{2+o(1)}$ simple operations.
Signature occupies $b^{1+o(1)}$ bits.

Users have cost constraints.

Cryptographers, cryptanalysts, implementors, etc. tend to focus on RSA and ECC, citing these cost constraints.

But we think that the most efficient unbroken *post-quantum* systems will be hash-based systems, code-based systems, lattice-based systems, multivariate-quadratic systems.